Lambda Expressions and Partial Evaluation Solutions

Copyright © James Raynard 2021

Partial evaluation

- Explain briefly what is meant by "partial evaluation"
  - Instead of processing all the data at once, the data is processed in stages
- Why is it useful?
  - It allows data to be processed when it becomes available or when it is needed
  - Reduces the amount of computation and simplifies processing

Copyright © James Raynard 2021

Storing Lambdas

- What type of variable is used to store a lambda?
  - The variable must have the same type as the functor generated by the compiler
  - Since this is not known to us, we must use auto
- Rewrite the program from the last exercise in which find_if was called with a predicate lambda that performed a capture by value
- This time, write the lambda separately, then store it and use the stored lambda as the predicate

Copyright © James Raynard 2021

Returning a Lambda from a Function call

- Write a function that takes a string and returns a lambda expression
- The lambda will take another string as its argument
- It combines this string with the function's argument string and returns the result
- Write a program which calls your function and stores the resulting lambda expression
- Call the stored lambda expression and display the result

Copyright © James Raynard 2021

Partial Evaluation using Lambda

- Here is some code from the lecture

```
auto greeter(const string& salutation) {
   return [salutation] (const string& name) { return salutation + ", "s + name; };
}

auto greet = greeter("Hello"s);
```

- Explain briefly how this code could be used to perform a partial evaluation
  - When we call greeter(), it processes the salutation part of the greeting, but not the name
  - To issue a greeting, we call greet with the person's name as argument

Copyright © James Raynard 2021

Capture by Reference Caveats

- Can storing a lambda that performs capture by reference be dangerous?
  - Yes
  - If the stored lambda is called after the captured variable has been destroyed, it will have a "dangling reference" to the variable
  - The program will probably crash
  - When calling this lambda, always check that the reference is still valid

Copyright © James Raynard 2021